Entrust TECHNOLOGIES

We Bring Trust to e-Business™

# Secure Roaming with Software Tokens

Michael J. Wiener

2000 September 14

# Non-Roaming Scenario

- User's private keys stored on local machine
  - private keys encrypted with password
  - called a software token
- Attack requires two steps
  - access to local machine to steal a copy of user's software token
  - correct guess of user's password

# Password Search

- Assume attacker steals a software token
  - we are not worried about a human typing in password guesses one at a time
- Sophisticated attacker:
  - automated candidate password generation
  - uses a dictionary of words, modified words, etc.
  - attempt to decrypt software token with millions of password guesses per day on many machines

# Moving to Roaming

- Protection from the fact that an attacker must penetrate local machine is gone
- Users want to be able to walk up to a computer they have never used before and log in
  - only means of user authentication is a password (for software-based solutions)
    - alternative is hardware tokens (expensive)

# Trivial Roaming (Insecure)

- All software tokens could be stored on a server with no access control
- User requests a software token, then decrypts locally with password
- Attacker can also request software tokens, then mount a password search attack
  - unacceptable risk

# Eliminating Password Search

- Software token must be delivered to user in strongly-encrypted form
  - software token = $E_{pwd}$(private keys)
  - send $E_K$(software token) to user
  - K is a strong, random key (>= 128 bits)
  - different K for each user
- Problem now is to deliver K to roaming user securely

# A Secure Solution

- SPEKE protocol
  - Simple Password Exponential Key Exchange
  - based on Diffie-Hellman key exchange
  - similar to other authenticated key exchanges
  - added features:
    - user authentication by password only
    - eliminates off-line password search by attacker

# SPEKE

**Client** **Login Server**

user name, $h(pwd)^x$ →

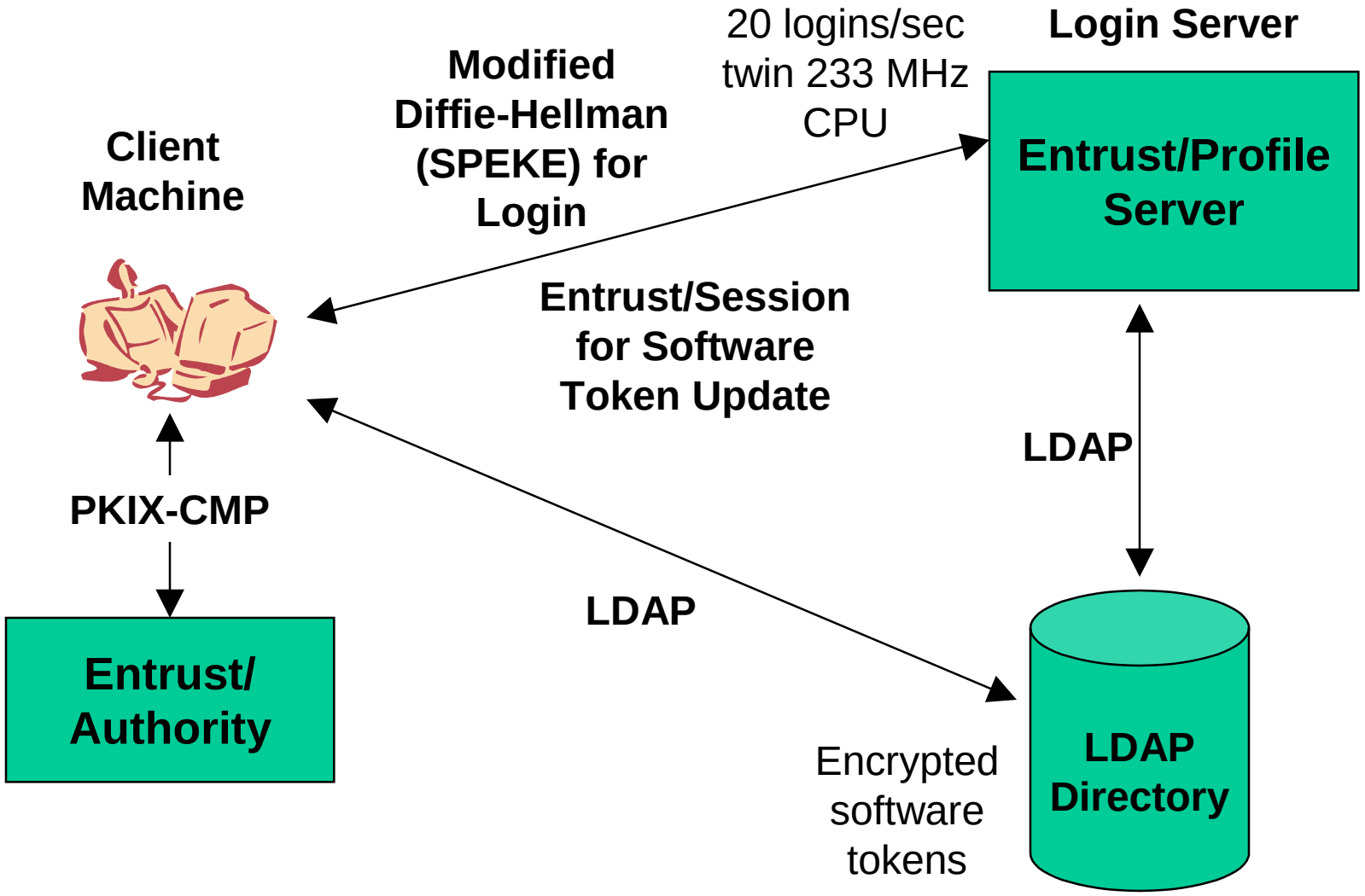
← $h(pwd)^y$, $E_{DHkey}(K)$, $E_K$(software token)

- Hash of password used as generator for Diffie-Hellman
- DH derived key used to encrypt strong key K

# SPEKE Security

- Attacker approaches
  - eavesdropping
  - impersonate user
  - impersonate server
  - man-in-the-middle
- In all cases, attacker cannot get any data to allow an off-line password search

# Entrust Roaming General Architecture



# Entrust Roaming Services

- User software tokens
  - stored centrally strongly-encrypted with key K
- Login server
  - authenticates user, delivers K securely
  - SPEKE protocol used
    - nothing in exchange allows attacker to test whether a password guess is correct
- User's machine
  - decrypts software token with K
  - decrypts private keys with password

# Alternative Approaches

- Multiple Servers
  - Kaliski/Ford paper
  - Jablon (inventor of SPEKE) has a protocol of this type as well
- “Virtual Smart Cards”
  - Hoover/Kausik paper

# Multiple-Server Approach

- User must contact 2 or more servers to log in
  - intended to reduce exposure if only one server is compromised
- Allows vendor to run one of the servers
  - creates possibility of service revenue on top of license revenue
- Challenges
  - log in performance issues
  - will connection to remote server be up 24x7 to allow logins?

# Virtual Smart Cards

- A type of software token designed so that wrong passwords give a valid-looking private key
  - only correct password gives correct private key
- User's key pair cannot be used for most normal PKI functions
  - doing so would undermine security

# Virtual Smart Cards Cont'd

- To prevent password searches, the following must remain secret
  - user public keys
  - data encrypted with user's public key
  - signatures
- Users can only interact with highly-trusted servers

# Summary

- Hardware tokens give a secure, but expensive solution for roaming
- Entrust’s software-only roaming solution gives good security at low cost